Association for Information Systems
AIS Electronic Library (AISeL)

ICIS 1987 Proceedings

International Conference on Information Systems (ICIS)

1987

AUTOMATED KNOWLEDGE ACQUISITION: OVERCOMING THE EXPERT SYSTEM BOTTLENECK

David Perry Greene
Carnegie Mellon University

AUTOMATED KNOWLEDGE ACQUISITION: OVERCOMING THE EXPERT SYSTEM BOTTLENECK

David Perry Greene
The Graduate School of Industrial Administration
Carnegie Mellon University

ABSTRACT

The artificial intelligence (AI) discipline of machine learning offers the best opportunity for alleviating the critical problem of acquiring the knowledge base necessary for expert systems. This paper examines the characteristics of such tasks and identifies a number of weaknesses with several dominant AI approaches. Genetic algorithms (GAs) are a probabilistic search technique based on the adaptive efficiency of natural organisms and offer an alternative which addresses the weaknesses in conventional methods. This paper describes the implementation of ADAM, a GA driven classifier, and compares the quality of the rules it generates to those of alternative induction techniques on a simulated decision problem.

INTRODUCTION

The process of eliciting knowledge from an expert is costly, time-consuming and prone to error (Hayes-Roth, Waterman and Lenat 1983; Michalski, Carbonell and Mitchell 1984). An effective alternative to this process is to build a knowledge base by providing examples of experts' decisions and allowing the system to determine the general rules. Furthermore, modelling what experts do rather than what they say overcomes the "paradox of expertise"[1] (Johnson 1983). However if such methods are to prove practical they must function under realistic problem conditions and requirements. Therefore this research will concentrate on the ability of learning methods to generate high performance rules from examples, under varying conditions, using limited prior knowledge in a format that is cognitively compatible with users.

"The ability to classify objects...is the basis for all inferential capacity" (Fisher and Langley 1985). At its simplest, the induction of rules from examples can be viewed as a classification problem (Rendell 1986; Holland et al. 1986). The learning task is one of finding the appropriate combination of features which partition a given set of objects into desired classes. Classifying the type of disease from a set of symptoms or the desirability of a stock investment from key indicators are examples of this task. Induction would occur as the system constructs more general classification rules to account for specific examples of an expert's decisions.

The quality of a learning system is measured by the effectiveness and functionality of the rules it generates. Specifically, a rule must classify as the expert would, it must be understandable and it should be easily implemented in a knowledge base. Effectiveness will be measured by comparing the classification a rule makes with those of the expert. The functionality of a rule will only be discussed briefly in the context of the representation used for the rules.[2]

The two dominant AI paradigms for the induction of classification rules are decision theoretic and symbolic concept acquisition (Michalski 1986). The assumptions and characteristics of these approaches will be shown to have a strong influence on the quality of the rules they generate. In comparisons of effectiveness versus functionality, these approaches tend to favor one dimension at the expense of the other. Genetic algorithms (GAs), because of their unique search mechanism, offer strengths on both dimensions and will be the focus of this research. While representational functionality is noted, the critical focus will be effectiveness. The hypothesis is that GAs will be able to predict decisions as well as traditional statistical methods while offering the representational superiority of the symbolic concept approach.

The remainder of this paper is divided into five sections. The first section describes the characteristics of the learning task. The second section examines the strengths and weaknesses in the two dominant paradigms. In the third section, an alternative approach using a genetic algorithm is developed to address the limitations in the existing methods. The fourth section describes a simulated problem to compare the three techniques. The fifth section presents the results followed by a brief discussion.

TASK CHARACTERISTICS

The most commonly used method of representing expert knowledge is in the form of condition-action pairs (or production rules) (Waterman 1986; Hayes-Roth, Waterman and Lenat 1983):

IF <CONDITION> THEN <ACTION> {1}

The "condition" typically is a conjunction of binary statements and the action may connote an actual procedure or the assignment of some value. In medical diagnosis, the condition might consist of a number of symptoms and the action would specify a disease or request for further tests; for financial investments, the condition could be characteristics of a stock and the action would be whether or not to purchase.

The binary statement in the condition can be described as a string of attribute-value (A-V) pairs called "selectors" (Michalski and Chilausky 1980). A specific stock could be represented in the following form:

[price=$20][industry=oil_and_gas][dividend=no] {2}

The conjunction of these A-V pairs form a "term" and the logical union of several terms forms a disjunctive expression or disjunctive normal form (DNF). DNF can also be considered a "compensatory" form since one term can compensate for another. An example of a compensatory rule in DNF for selecting a stock portfolio would be:

IF [price < $40][PE_ratio < 15%] {3}
[industry=oil_and_gas][dividend=yes]

OR [price < $20][industry=technology]
[PE_ratio < 25%][company_age > 5yrs]

THEN purchase

A conjunctive rule form would consist of a single term (no or's). A "training set" would consist of a number of previously classified examples in the form of a vector of A-V pairs such as in expression {2} plus the expert's classification. The learning task would be to develop a rule that can discriminate positive examples of a concept (e.g., purchaseable stocks) from the negative.

Task Complexity

A number of factors cause this to be a difficult problem. As Valient (1985) indicates, learning disjunctions of conjunctions is computationally complex for a reasonable number of features and becomes NP-hard in certain circumstances. To see this, using only 40 binary selectors (dummy variables), the search space for a single term is the 2^{40} possible combinations. However, an individual is quite likely to have rules which use multiple terms ("term 1" *or* "term 2" *or* ..."term n") . For simplicity, if we restrict this compensatory decision rule to only 5 terms, the number of possible rules would be, $2^{40} \times 2^{39} \times 2^{38} \times 2^{37} \times 2^{36}$ or approximately 10^{58} possible combinations.[3]

Another aspect of complexity in interpreting examples is the difficulty in unambiguously characterizing a person's decision based on his actions (Newell and Simon 1972; Einhorn 1970; Waterman 1986). Furthermore, errors (or "noise") may arise between a decision strategy and a classification and between the classification made versus the one recorded. Quinlan (1986) describes noise as primarily affecting the formation and the use of the discovered rules. For symbolic concept techniques, which build rules in a stepwise fashion, the effects of noise can be severe.

CONVENTIONAL APPROACHES

Two dominant approaches to induction are decision theoretic (or statistical pattern recognition) and symbolic concept acquisition (SCA). The key difference is that the traditional statistical pattern recognition methods use continuously changeable parameters to express discriminating boundaries and have a strong numerical flavor while SCA learns to describe a concept through the manipulation of symbolic representations (Rendell 1986; Michalski 1986).

Statistical Pattern Recognition

Traditional statistical models such as discriminant analysis or regression models attempt to represent a decision rule as a combination of attribute weights associated with each A-V pair used in the training example. All the attributes are numerically coded and the search for parameter weights is conducted through mathematical manipulations of means, frequencies and variances. The evaluation function that directs the search is based on a measure such as mean squared error, Bayes Theorem or maximum likelihood estimates. An object X is a member of class Y if the attribute values of $X=(x_1,x_2...,x_k)$ with weights $W=(w_1,w_2...,w_k)$ result in $w_1x_1 + w_2x_2 +...+ w_kx_k > y$ where y is the threshold of class Y.

The strength of traditional statistical methods is their "robustness" or effectiveness across a wide range of problem conditions (Dawes and Corrigan 1974; Dawes 1979). Evidence of this comes from a large body of research where statistically generated rules outperformed the experts they were modelling (Dawes 1979; Slovic 1969). However, most traditional methods assume continuous tradeoffs (implicit disjunctions) among attributes, a distribution which frequently does not hold among real-world problems (Cohen and Feigenbaum 1982). While violating this assumption is often non-problematic, under some conditions it can generate serious errors (Valient 1985; Curry, Louviere and Augustine 1981; Johnson, Meyer and Ghose 1985). Furthermore, traditional statistical methods are prone to difficulties with small samples (Nilsson 1969). A more serious criticism is that representing rules as numeric coefficients provides little intuition or understanding (Cohen and Feigenbaum 1982; Fisher and Langley 1985). This is critical because most expert systems require rules that are comprehensible to both the experts and users (Hayes-Roth, Waterman and Lenat 1983; Waterman 1986).

Symbolic Concept Acquisition

A prototypical example of SCA is Quinlan's ID3 (1984; 1986) classification system. The algorithm is primarily a search tree which sequentially builds systems of production rules by considering the discriminability of selectors in the training examples. For example, given a list of stocks, their attributes coded as selectors (such as price<$20, PE_ratio>15), and an expert's judgement (buy or not-buy), ID3 searches by evaluating each selector for its ability to discriminate between the expert's decisions using an information-theoretic measure. The selector which provides the best separation (or fit) becomes a "branch" of the search tree. This process continues iteratively until all the stocks are separated into the buy or don't buy class. The rule would then consists of all the selectors used to build the tree as in expression (3).

The use of a production system formalism is an important advantage. Because of their apparent consistency with human thinking (Newell and Simon 1972; Klahr, Langley and Neches 1987) production system representations are easily understood by users, which is necessary if the rules are to be accepted (Waterman 1986). Another feature is modularity which allows knowledge, as individual rules, to be easily added or removed from a knowledge base (Newell and Simon 1972; Cohen and Feigenbaum 1982). Unfortunately the use of production rules also exposes the complexity of DNF problems discussed earlier. As a best-fitting algorithm, rules from the tree building approach become complex and inaccurate as attempts are made to explain noise.[4] Another concern is that the decision tree procedure is locally optimal but not necessarily globally optimal (Breiman et al. 1984), which means constructing a rule piece by piece will be ineffective if it is critical *combinations* of pieces which provide superior performance. Therefore, when conditions are less than ideal, the predictive quality of the rule may fare poorly.

GENETIC ALGORITHMS AND ADAM

Genetic algorithms (GAs), developed by John Holland (1975), are a probabilistic search method based on the concept of adaptive efficiency in natural organisms. In nature, the members of a species which are best suited to the environment are the most likely to survive and produce offspring. Since the offspring are likely to inherit these survival traits, the succeeding generation will contain more fit individuals. If the environment can only support a limited population, then the standards of fitness rise and each successive generation should contain better individuals.

By representing a DNF production system rule, such as expression (3), as a string of binary selectors (a rule-string), the same search operation can be performed. To locate the best rule-strings, the algorithm creates a population of rules randomly and evaluates their "fitness" on some performance measure. Rules which score well possess above-

average traits or selector combinations. These good rules survive and breed better offspring, creating a new population of improved rules. This process continues until some performance level is achieved or a fixed number of generations have elapsed. Since the binary string is a direct encoding of the DNF rule, the advantages of using production systems are retained. However, as will be described, the *implicit parallelism* of GAs can overcome the combinatorial complexity.

The key features are selecting the rules to survive and providing operators for "breeding." Rule survival is a stochastic process with the probability determined by how much better a rule's score is than the average fitness score of that generation. If the population average is 2.5 and a certain rule scores 10, it should be four times more likely to be selected as a parent than other rules. Since sampling is with replacement and there are likely to be four times as many copies of that rule to act as parents, the next generation of rules is likely to have many new variations of the higher quality genes from that parent.

"Breeding" results from randomly pairing the selected parent rules and applying two genetic operators. The primary operator is *crossover* which chooses a random position for the two rules and swaps all the selectors to the left as follows (x_i and y_i represent selectors):

(1) $x_1\ x_2\ x_3\ x_4\ x_5\ x_6$ $\quad$ $y_1\ y_2\ y_3\ y_4\ y_5\ y_6$

(2) $x_1\ x_2\ x_3\ x_4\ \ x_5\ x_6$ $\quad$ $y_1\ y_2\ y_3\ y_4\ \ y_5\ y_6$

(3) $x_1\ x_2\ x_3\ x_4\ y_5\ y_6$

$y_1\ y_2\ y_3\ y_4\ x_5\ x_6$

This results in two new "child" rules made up of previously successful selector combinations from their "parents." The probability of crossover occurring for any parent pair is high but less than one so that some good rules enter the next generation intact. A second minor operator, occurring with low probability, is *mutation*, which randomly changes a value and assures that, in theory, no combination in the search space is unreachable.

These genetic operators provide the search mechanism necessary to locate the selector combinations required for the best rule. With a large search space, SCA methods resort to heuristics to decide which feature to add in piece-wise fashion leading to possibly sub-optimal and noise plagued rules (traditional statistical methods deal with the parameter weights, not the actual features). Instead of building a rule feature by feature, the GA evaluates a whole set of features as a complete (condition -> action) rule, for example:

if (A and C and not D) **or** (B and C and E)
then choose,[5]

where A through E are simple true/false conditions (e.g., A: price > $15). If a specific attribute were to dominate performance, that is, be the only critical feature in an expert's decision, then a rule which focused on that attribute *only*, would prove superior in predicting the expert's behavior (assuming for the moment that performance is measured by predictive accuracy and simplicity). This adaptation to essential features allows a GA to focus on a single attribute. On the other hand, it is not clear that the sequential search tree (as in ID3) can pick up on a pair of attributes that the expert considers important as a combination yet which are individually dominated by a less important but stronger single attribute.

The combinatorial problem of using "complete rules" is resolved by the "implicitly parallel" properties of the GA. "Implicit parallelism" results from testing *building-blocks* contained in each rule and throughout a *population* of rules, which are *recombined* to generate new rules to advance the search. The performance of any rule can be viewed as representing the total effect of all possible combinations of its features. For example, a hypothesized rule which says: IF A = true and B = true and C = false THEN purchase the stock, has multiple combinations of features. If the rule works better than average it could be due to all three conditions (A, B, and C) or due to A and C only, with B being irrelevant or even detrimental, or it might be some other combination. Each of these implicit combinations represents the *building-blocks* which could account for the rule's good performance; exactly which combination is best is not known, however the expectation is that the better building-blocks are in the above average rules. By using a *population* of potential rules, many variations can exist across the different building-block combinations. At each time step, all the rules are explicitly tested and therefore all the building-blocks, both within and across

the rules, are *implicitly* tested *in parallel.* By randomly *recombining* the features of better rules, a new population is created providing new variations of good building-blocks. Over successive iterations of this process, the population will begin to converge on the best combinations which represent the best rules. In this manner, the search for the best rules is done in an implicitly parallel fashion via the best building-blocks from a population of possible rules.

Holland (1975) provides the original theoretical analysis which proves GAs to be an efficient, if not optimal, sampling/search technique for large problem spaces. Moreover, because multiple rules are maintained and selection is probabilistic, the search does not fall prey to noise or minor inconsistencies. It should be noted that, as currently implemented, GA's conduct search quite slowly in terms of operating time, however, real-time performance is not necessarily an issue when acquiring rules for inclusion in an expert system. Furthermore, the time does not increase exponentially with the size of the problem. While GAs have been successfully applied to rule learning (for poker [Smith 1980] and gas pipeline operations [Goldberg 1983]), the training examples have been from environmental cues, not experts. One question is how well a GA can learn rules in DNF form under conditions which could occur when modelling an expert's decisions. A second question is how a GA will compare to more traditional approaches. To investigate these issues, a GA driven classifier called ADAM (for A Decision Acquisition Model) was developed and compared to a statistical Logit model and an implementation of ID3 called CLS[6] (Currim, Meyer and Le 1986) on a simulated induction problem. A brief description of ADAM can be found in the appendix; for a more detailed description see Greene and Smith (1987).

METHODOLOGY

A simulation was used rather than a real problem to control the environment for a better understanding of how different conditions would affect ADAM, CLS, and a linear Logit model. Since examples used in realistic induction problems are likely to be generated from different decision strategies and vary in levels of quality and completeness, the following four factors were investigated for their potential effect on performance:

1. Type of decision strategy (conjunctive, compensatory or mixed).
2. Number of attributes on which the rule is based (3, 6, 9).
3. Level of noise in representing the choice (0%, 10%, 20%).
4. Sample size used for estimation (20, 100, 200) with half as a holdout.

To provide the simulation data, a table of randomly generated A-V terms was created. Each alternative consists of three, six, or nine attributes represented by a random number between 0 and 99. A coding function using a decision maker's strategy (conjunctive, compensatory or mixed) was applied resulting in a set of decisions marked as "positive" or "negative" examples. This could be thought of as a sample of stocks characterized by three, six or nine features plus an indicator of whether or not the expert thought it should be purchased.

The choice indicator was generated using the following three choice functions:

conjunctive: decision =

1 if (X1 > t1) and (X2 > t1) and...(Xn > t1)

0 otherwise

compensatory: decision =

1 if (X1 + X2 + ... Xn)/ n > t2

0 otherwise

mixed: decision =

1 if ((X1 > t3) and (X2 + X3 + ... Xn)/n > t4)

0 otherwise

* n = number of attributes in a given experimental condition (n ϵ {3,6,9})

* X_i = a given attribute (i ϵ {1,2,...n })

* t_j = thresholds, each t will be selected a priori for each of n conditions to generate an approximately equal split between the number of "chosen" and "nonchosen" alternatives.

Noise was introduced into each coded set of examples by changing the decision indicator of any alternative in the set with a probability of 0%, 10% or 20%. This represents a severe form of misclassification since an alternative which contained acceptable attribute values is now indicated unacceptable and vice versa.

Using combinations of choice strategy, number of attributes, and noise level, nine selection models were created representing a 3 x 3 x 3 partial factorial. Each of the nine models is applied for three different sample sizes yielding 27 (model/sample sizes). Each of these 27 conditions is repeated five times, yielding 135 data sets. Half of each data set was used as a holdout sample meaning it represented a set of decisions not previously seen and therefore usable for prediction. ADAM, CLS and the simulation were all programmed in PASCAL and run on an IBM-PC. The Logit results were generated using Hotztrans on a VAX computer and with RATS on an IBM-PC.

RESULTS

The objective of the ADAM was to simulate performance under a number of conditions and to determine whether it offered any improvements with respect to the issues described in earlier sections.[7] The focus here is how effective are ADAM's rules compared to the other approaches. For the simulation, effectiveness is measured by how well the model's rule predicts the hold-out sample. The comparative predictive levels of the models averaged over the five repetitions are presented in Table 1.

It is evident that ADAM, using a genetic algorithm, generated rules with equal or superior predictive ability to those of CLS across almost all the experimental conditions (the exception being one and two points difference for the mixed model with nine attributes). In comparing ADAM to the Logit model, the major impression is how comparable and consistent their performance was. Overall, ADAM predicted with 80.7% accuracy versus Logit at 79.9%. As was expected, the performance varied across conditions, as shown in Table 2.

As seen from Table 2, ADAM appears to offer a slight edge with respect to conjunctive rules and small sample sizes, areas where traditional models do not perform as well. However, none of the performance differences were found significant.

With *percentage correct* as the dependent variable, a comparison of multiple regression models using dummy variables primarily examined main effects.[8] The results indicate significant differences ($p < .0001$) for all main effects (model type, number of attributes, %-noise, sample size) consistent with expectations. That is, increasing noise and larger attribute sets had detrimental effects on predictive accuracy, while larger samples had a positive effect. The performance of ADAM showed significant improvement over CLS; however, an F-test between regression models did not indicate significant difference over Logit at ($p < .05$), even with first order interactions included.

One explanation for the surprising strength of the Logit model on conjunctive rules was the nature of the simulation environment. As several researchers have noted (Dawes and Corrigan 1974; Curry, Louviere and Augustine 1981; Johnson, Meyer and Ghose 1985), the use of a uniform distribution in generating simulation attributes provides a best case environment for the averaging of a statistical model. However, such distributions are unlikely to occur in a real-world environment (Cohen and Feigenbaum 1982; Curry, Louviere and Augustine 1981). A modification to the simulation is currently under way.

Two encouraging findings were the low variance of ADAM's results across repetitions of the trials and the stability of ADAM across differences in both strategy and noise, supporting the expectations for genetic search. The falloff in prediction is consistent with the increase in the noise level. Several additional runs using noise as the only experimental variable support this finding. The slightly lower performance in compensatory rules may be attributable to the loss of information caused by encoding a 100 value random number as a dichotomous variable. Modifications for this effect will be investigated in future research.

When experts describe their rules, they frequently do not place the same weight on all the features but instead indicate that certain observations are more important than others (Michalski and Chilausky 1980; Waterman 1986). In a regression model such as Logit, this is represented by the beta coefficients or parameter weights. A comparable parameter is ADAM's relative frequency measures for each selector. Since this measure is one of the strengths of a regression, even under the simulation conditions, it would be interesting to compare how

Table 1. Effectiveness of ADAM, CLS and Logit
(percentage of holdout cases correctly predicted)

				sample = 10			sample = 50			sample = 100		
	Strategy	Atrib	Noise	ADAM	CLS	Logit	ADAM	CLS	Logit	ADAM	CLS	Logit
1	**Conj**	**3**	**0%**	100	90	92	100	100	100	100	100	100
2	**Conj**	**6**	**10%**	72	62	60	73	67	75	76	58	72
3	**Conj**	**9**	**20%**	86	73	66	80	76	80	78	66	76
4	**Comp**	**3**	**10%**	76	72	73	79	65	76	83	69	80
5	**Comp**	**6**	**0%**	75	59	83	82	68	84	78	66	82
6	**Comp**	**9**	**20%**	62	47	71	66	64	70	66	56	71
7	**Mixd**	**3**	**20%**	78	68	73	76	69	73	75	66	78
8	**Mixd**	**6**	**10%**	88	47	84	88	80	86	81	80	85
9	**Mixd**	**9**	**0%**	78	74	77	92	90	95	87	88	94

Table 2. Performance Across Conditions

		MODEL			NOISE			SELECTORS			SIZE		
	overall	conj	comp	mixd	0%	10%	20%	3	6	9	10	50	100
ADAM	80.7	85.0	73.8	82.5	87.7	82.1	71.5	85.2	78.9	77.2	79.0	82.0	81.2
Logit	79.9	80.1	76.7	83.0	89.6	78.9	71.7	83.0	79.0	77.8	75.4	82.3	82.0

closely ADAM and Logit (as the standard) weight attribute importance. A critical question is whether production rules can provide diagnostic validity.

As is evidenced in Table 3, not only do the two algorithms generate equivalent predictions, but they also appear to agree as to the relative importance of attributes even across sample sizes. The correlations appear to follow increasing convergence as sample size increases, although this trend was not significant ($p < .01$). The results lend support to the use of production rule models in providing useful quantitative measures even while representing a data set symbolically.

Based on a simple simulation, the effectiveness of ADAM was evaluated in comparison with two, more traditional, methods. With respect to the research objectives and performance hypotheses, the GA

Table 3. Diagnostic Correlation between ADAM and Logit (Correlation between Relative Frequency of Attributes in ADAM and Beta Coefficients from Logit)

		Sample Set Size		
Attrib	(cases)	10	50	100
X1	(135)	0.45a	0.83a	0.78a
X2	(135)	0.50a	0.74a	0.79a
X3	(135)	0.41b	0.75a	0.75a
X4	(90)	0.64a	0.65a	0.76a
X5	(90)	0.48b	0.62a	0.65a
X6	(90)	0.65a	0.58a	0.65a
X7	(45)	0.43	0.85a	0.61c
X8	(45)	0.47	0.48	0.42
X9	(45)	0.53d	0.56c	0.72b

significance

a- $p < .0001$

b- $p < .001$

c- $p < .01$

d- $p < .05$

provided equal or superior performance to CLS across all measures, especially in those areas addressing the weakness of symbolic concept models. Further, the GA performed very well with respect to the traditional strengths of the statistical model while providing the important benefits of production system representation. Accepting that the simulation represents a simplified situation, overall, the results appear to provide strong support for the potential of genetic search as a method for modelling decision rules in a knowledge acquisition task.

DISCUSSION

Automating knowledge acquisition through inductive classification algorithms offers a way of overcoming the bottleneck in expert systems. To be worthwhile, such methods must generate effective and functional rules acquired from examples under inhospitable conditions. Two dominant learning paradigms were shown to have contrasting weaknesses which a genetic algorithm might overcome. A classification system called ADAM was developed utilizing a GA for search. For the purposes of the paper, a simplified decision simulation was used to evaluate GA performance and to provide a comparison to the earlier methods. The results of the simulation support the potential of both genetic search and the use of ADAM for knowledge acquisition.

Production systems appear to offer an advantage over traditional statistical coefficients and genetic search appears more robust than a prototypical heuristic method. Several issues need to be examined. With respect to the problem domain, it is important to look at increasing the number and type of attributes as well as different distributions of attribute sets. In addition, recent improvements to induction trees as well as other SCA methods may yield better performance. With respect to the algorithm, as noted, the current representation potentially loses information so that exploration to allow ADAM to modify its coding could prove worthwhile. An important next step will be to apply an upgraded ADAM to an actual induction problem in a complex domain, possibly medical diagnosis. Overall, the positive results suggest a

much more detailed investigation of the genetic model for acquiring expert knowledge is warranted.

ACKNOWLEDGMENT

The author wishes to thank Bob Meyer, Steve Smith, Eric Johnson and Mike Prietula for their invaluable help during the evolution of this project.

ENDNOTES

[1] The finding that as experts become more competent they are less able to describe their real behavior. Another consideration is the seminal work of Nisbett and Wilson (1977) on the limits of verbal self-reports.

[2] An operational measure of functionality was not developed; however, it is argued that an identical production-system representation, used by both the GA and the symbolic approach, is superior to the numerical-coefficient representation used by traditional statistical methods.

[3] The possible combinations equal $(2^k)!/(2^{k-t})!$, where k = (the number of selectors) and t = (the number of terms in a rule).

[4] Quinlan (1986) offers some possibilities which could remedy noise and aid tree pruning; however, these were not able to be implemented in time and will have to be evaluated in a later study.

[5] This encoding of DNF is possible in a GA by adapting the flexible representation originally developed by Smith (1980).

[6] This *CLS* is not Hunt's 1966 approach (Michalski, Carbonell and Mitchell 1984), but an implementation of Quinlan's ID3 using Quinlan's (1984) information theoretic entropy measure. The CLS code was supplied by Bob Meyer at the University of California, Los Angeles.

[7] If the use of production systems is accepted as superior to numeric coefficients in terms of functionality (comprehensible, modular), then the representation of ADAM is equal to CLS and superior to Logit.

[8] Because of the fractional factorial design, the available statistical packages were unable to adequately handle an ANOVA except at the aggregate, "main effects," level. Therefore a series of *multiple regressions* were done in which the various possible interactions were incrementally removed to evaluate their influence on the dependent variable (prediction %).

REFERENCES

Breiman, L.; Friedman, J.; Olshen, R.; and Stone, C. *Classification and Regression Trees.* Wadsworth Inc., Belmont, CA, 1984.

Cohen, P. R., and Feigenbaum, E. A. *The Handbook of Artificial Intelligence.* Volume II, William Kaufmann, Los Altos, CA, 1982.

Currim, I.; Meyer, R. J.; and Le, N. "A Concept-Learning System for the Inference of Production Models of Consumer Choice." Working Paper Number 149, Center for Marketing Studies, University of California, Los Angeles, 1986.

Curry, D. J.; Louviere, J. J.; and Augustine, M. J. "On the Sensitivity of Brand-Choice Simulations to Attribute Importance Weights." *Decision Sciences*, Vol. 12, 1981, pp. 502-516.

Dawes, R. M. "The Robust Beauty of Improper Linear Models in Decision Making." *American Psychologist*, Vol. 34, No. 7, July 1979, pp. 571-582.

Dawes, R. M., and Corrigan, B. "Linear Models in Decision Making." *Psychological Bulletin*, Vol. 81, No. 2, 1974, pp. 95-106.

DeJong, K. A. *Analysis of the Behavior of a Class of Genetic Adaptive Systems.* Unpublished Ph.D. Thesis, Department of Computer and Communication Sciences, University of Michigan, 1975.

Einhorn, H. J. "The Use of Nonlinear, Noncompensatory Models in Decision Making." *Psychological Bulletin*, Vol. 73, No. 3, 1970, pp. 221-230.

Fisher, D., and Langley, P. "Approaches to Conceptual Clustering." *Proceedings of the Ninth International Joint Conference on Artificial Intelligence.* (Los Angeles, California) Morgan Kaufmann, 1985, pp. 691-697.

Greene, D. P., and Smith, S. F. "A Genetic System for Learning Models of Consumer Choice." *Proceedings of the Second International Conference on*

Genetic Algorithms and their Applications, Boston, Massachusetts, 1987.

Goldberg, D. *Computer Aided Gas Pipeline Operation Using Genetic Adaptive Systems.* Unpublished Ph.D. Thesis, Department of Civil Engineering, University of Michigan, 1983.

Hayes-Roth, F.; Waterman, D. A.; and Lenat, D. *Building Expert Systems.* Addison-Wesley, Reading, MA, 1983.

Holland, J. H. *Adaptation in Natural and Artificial Systems.* University of Michigan Press, 1975.

Holland, J. H. "Escaping Brittleness: The Possibilities of General Purpose Learning Algorithms Applied to Parallel Rule-Based Systems." In R. Michalski, J. Carbonell, and T. Mitchell (eds.), *Machine Learning: An Artificial Intelligence Approach*, Volume II, Morgan Kaufmann, Los Altos, CA, 1986.

Holland, J. H.; Holyoak, K. J.; Nisbett, R. E.; and Thagard, P. R. *Induction: Processes of Inference Learning and Discovery.* MIT Press, Cambridge, MA, 1986.

Johnson, E. J.; Meyer, R. J.; and Ghose, S. "When Choice Models Fail: Compensatory Models in Efficient Sets." Working Paper, Graduate School of Industrial Administration, Carnegie Mellon University, 1985.

Johnson, P. E. "What Kind of an Expert Should a System Be?" *The Journal of Medicine and Philosophy*, Vol. 8, 1983, pp.77-97.

Klahr, D.; Langley, P.; and Neches, R. *Production System Models of Learning and Development.* MIT Press, Cambridge, MA, 1987.

Michalski, R. "Understanding the Nature of Learning." In R. Michalski, J. Carbonell, and T. Mitchell (eds.), *Machine Learning: An Artificial Intelligence Approach*, Volume II, Morgan Kaufmann, Los Altos, CA, 1986.

Michalski, R., and Chilausky, R. L. "Knowledge Acquisition by Encoding Expert Rules Versus Computer Induction from Examples: A Case Study Involving Soybean Pathology." *International Journal of Man-Machine Studies*, Vol. 12, 1980, pp. 63-87.

Michalski, R.; Carbonell, J.; and Mitchell, T. "An Overview of Machine Learning." In R. Michalski, J. Carbonell and T. Mitchell (eds.), *Machine Learning: An Artificial Intelligence Approach*, Volume I, Tioga Press, 1984.

Newell, A., and Simon, H. *Human Problem Solving.* Prentice Hall, Englewood Cliffs, NJ, 1972.

Nisbett, R. E., and Wilson, T. D. "Telling More Than We Can Know: Verbal Reports on Mental Processes," *Psychological Review*, Vol. 84, No. 3, May 1977, pp. 231-259.

Nilsson, N. J. *Learning Machines.* McGraw-Hill, New York, 1969.

Quinlan, J. R. "Inductive Inference as a Tool for the Construction of High-Performance Programs." In R. Michalski, J. Carbonell and T. Mitchell (eds.), *Machine Learning: An Artificial Intelligence Approach*, Volume I, Tioga Press, 1984.

Quinlan, J. R. "The Effect of Noise on Concept Learning." In R. Michalski, J. Carbonell, and T. Mitchell (eds.), *Machine Learning: An Artificial Intelligence Approach*, Volume II, Morgan Kaufmann, Los Altos, CA, 1986.

Rendell, L. "A General Framework for Induction and a Study of Selective Induction." *Machine Learning*, Vol. 1, No. 2, 1986, pp. 177-226.

Slovic, P. "Analyzing the Expert Judge: A Descriptive Study of a Stockbroker's Decision Processes." *Journal of Applied Psychology*, Vol. 53, No. 4, August 1969, pp. 255-263.

Smith, S. F. *A Learning System Based on Genetic Adaptive Algorithms.* Unpublished Ph.D. Thesis, Department of Computer Science, University of Pittsburgh, December, 1980.

Smith, S. F. "Adaptive Learning Systems." In R. Forsyth (ed.), *Expert Systems, Principles and Case Studies*, Chapman and Hall, Ltd., 1984, Chapter 11.

Valient, L. G. "Learning Disjunctions of Conjunctions." *Proceedings of the Ninth International Joint Conference on Artificial Intelligence* (Los Angeles, California) Morgan Kaufmann, 1985, pp. 560-566.

Waterman, D. A. *A Guide to Expert Systems.* Addison-Wesley, Reading, MA, 1986.

APPENDIX

This section will briefly describe some of the more specific details of ADAM. Some familiarity with the relevant features of genetic algorithms is assumed. The description is divided into three sections: the representation, the evaluation, and the rule generation. For a more detailed description see Greene and Smith (1987).

The representation used is a simple coding over the alphabet {0,1,#} (# representing a don't-care position), of each selector into a *term* expressed as a string of length *n* where *n* equals the number of dichotomous selectors defined for the simulation (three, six, or nine). A *complete rule* is then a concatenation of one or more terms allowing implicit disjunctive normal form, that is, within a single term an *and* relation among selectors is assumed and between terms an *or* relation is assumed. The decision state is indicated positive if the conditions of any of the terms match the conditions in an example.

The evaluation function is a weighted summation of three measures: prediction, specificity and term-count. Prediction is a simple match score of the number of times a rule was activated by a positive example and not activated by a negative example over the total number of examples. Specificity measures the number of don't-care positions (#) over the total length of the rule string. The assumption is, ceteris paribus, rules with fewer defined positions can be applied in more situations and should be favored. Term-count primarily helps provide bias among rule structures. Prediction is always the dominant component but the weighting between the other measures is allowed to shift based on population characteristics to provide a necessary discrimination in the latter stages of the search.

Rule-generation is based on a modified crossover, using a population of 50 strings initially generated at random. The population is replaced each generation with the incorporation of an "elitist" strategy (DeJong 1975). The probability of crossover is set at 0.6. Crossover operates between terms and between selectors as outlined by Smith (1980) with a modification permitting single terms to occur and be included in the crossover process. Mutation was set at 0.001.